Distribution System Topology Detection Using Consumer Load and Line Flow Measurements

Raffi Avo Sevlian and Ram Rajagopal

***Abstract*—We present a topology detection method combining smart meter sensor information and sparse line measurements. The problem is formulated as a spanning tree identification problem over a graph given partial nodal and edge power flow information. In the deterministic case of known nodal power consumption and edge power flow we provide sensor placement criterion which guarantees correct identification of all spanning trees. We then present a detection method which is polynomial in complexity to the size of the graph. In the stochastic case where loads are given by forecasts derived from delayed smart meter data, we provide a combinatorial complexity MAP detector and a polynomial complexity approximate MAP detector which is shown to work near optimum in all numerical cases.**

I. Introduction

The need for advanced controls in the distribution system is an emerging topic in power system and the controls community. Proposed computational models for problems such as dispatching of distributed energy resources [1], [2] or coordinated voltage control [3], [4], [5], [6] assume known system topology and network parameters. In reality customer level feeders are not known to within the accuracy as that of the transmission system. Therefore, situational awareness at the distribution system level is needed for future smart grid applications.

The classical power systems state estimation approach has has been adopted by a number of authors. Such a method relies on the Generalized State Estimator (GES) [7], [8] which will solve for system voltages and breaker statuses. In a GES, breaker status variables are included into the state estimation equations as $\{0, 1\}$ variables that are usually relaxed and solved by rounding strategy [9]. These methods have been applied to radial distribution system with changing topology as well. There is recent interest around distribution systems topology detection. In [10], a traditional GSE is employed in identifying the correct topology in a distribution system. They run a traditional WLS state estimator and use dummy variables for breaker status indicators.

Many non-conventional methods provide new insights into the estimation and detection problems. In [11] the authors introduce the use of high frequency micro-Phasor Measurement Unit (μPMU) data in the topology detection task. They propose a method of comparing simulation and measured μPMU data for each topology. In [12] the authors develop a voltage time series approach to identifying topology changes relying on high accuracy μPMU data. In [13], [14] the authors develop a method using nodal voltage analysis to reconstruct the grid topology in real time. Similarly, [15] presents a method of reconstructing the network based on voltage correlation analysis.

We should note some important features of each work. In [10], [11] no approximations are made in measurements or system models and make no use of the tree structure of the distribution system. In [10] all forms of measurement are considered since they rely on a general model formulation. In [11], high frequency voltage magnitude and phase measurements are assumed at each bus. In [12], [13], [14] structural properties of the network are exploited, and linear models relating true power injections and observed voltage magnitude/phase are used. Additionally none of the method are used to provide a sensor placement strategy separate from heuristic evaluation of various scenarios. In [15], the authors provide a method similar to [13], [14] where correlation analysis is used. However they rely on long time captures, so their method is more in line of network discovery not real time topology detection. In [16] the authors present a general state estimator based method that is used in topology detection. They use the method for a heuristic placement strategy.

Our work differs from these more traditional methods for a number of reasons. We focus on a simplified model measuring only power flow in a network. Our model does not consider voltage measurements or losses in the network. Second we make the assumptions similar to [13], [14] where we aim to detect the operational tree structure of the network. We propose a general *Spanning Tree Identification Problem* and show the necessary and sufficient conditions for identifiability of all spanning trees under the measurement and power flow models.

The paper is organized as follows. Section II formulates the problem of topology detection. We present the load and sensor models used in the identification problem along with the sensor placement optimization. Sections **??** and V solve the detection and sensor placement problems in the deterministic and stochastic cases respectively. Finally, numerical demonstrations are given in Section VI.

II. Problem Formulation

The topology detection problem is formulated as a *spanning tree identification* problem. Consider a power distribution network that has a planar graph structure but is operated as a

R. Sevlian is with the Department of Electrical Engineering and the Stanford Sustainable Systems Lab, Department of Civil and Environmental Engineering, Stanford University, CA, 94305. Email: rsevlian@stanford.edu.

R. Rajagopal is with the Stanford Sustainable Systems Lab, Department of Civil and Environmental Engineering, Stanford University, CA, 94305. R. Rajagopal is supported by the Powell Foundation Fellowship. Email: ramr@stanford.edu.

spanning tree at a given moment in time. Power is supplied from the substation to a set of feeders which are the at the root of multiple loads. In network reconfiguration, sets of breakers and tie switches can reconfigure themselves such that all loads are connected and no feeders are connected.

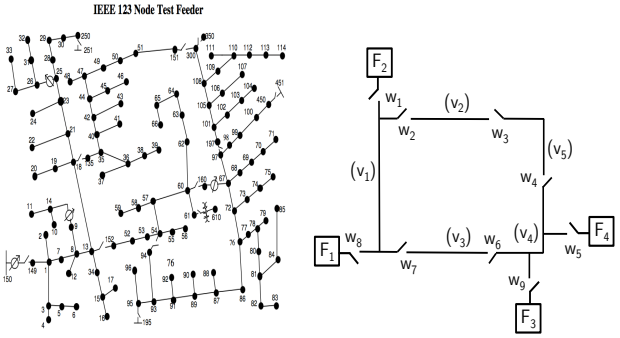
In the detection problem, we assume the breaker/tie switching statuses are not available and must be inferred from the available power flow information. This is of practical concern since generally only feeder head breakers are automated, while other devices are mechanically set and reset by work crews. A usual concern in such situation is removing human error by automatically detecting the status of these devices from information currently deployed on the network. We assume the following information is available:

1) Load pseudo-measurements in the form of forecasts which have forecast errors that must be taken into account. This information is available from Smart Metering Infrastructure.
2) Real-time measurements of the power flows on a fraction of the lines obtained using noiseless sensors placed on the selected lines. These are modeled as noiseless since the errors of these measurements are negligible compared to those of the load pseudo-measurements.

The detection and placement problem is solved in two scenarios: (1) *Deterministic Case*: loads and flows are known perfectly, (2) *Stochastic Case*: loads are known with uncertainty due to forecasting error.

III. System Model

A. Island Graph



(a) (b)

Fig. 1. (b) Typical test feeder with sectionalizing switch operation. (b) Island graph simplifying topology of feeder.

Consider, for example, the IEEE 123 node feeder in Figure 1(a). There are multiple loads connected to each other and separated by switches. We can then think of the network as a set of connected islands, separated by various switches. The typical feeder will have multiple power sources i.e. feeders. The set of switches are appropriately set so that each group of islands is connected to a load while ensuring no flow from one feeder to another. Therefore the distribution system must always maintain a tree structure.

This network can be reduced to a simplified graph as shown in Figure 1(b). In this *island graph*, islanded loads are lumped together, while source of power are represented as feeder nodes. For example, sources 150, 251, 195 and 451 in Figure 1(a) are Feeder nodes F_1, F_2, F_3, F_4 in 1(b). Additionally, the set of loads connected to these feeders, and each other are converted to loads $v_1, \ldots, v_4$ in Figure 1(b).

In the island graph, we must maintain a switch configuration such that no feeders are connected to each other. The connection of two feeders, for example by both w_1 and w_8 being closed will violate the tree structure of the network since two feeders will be energizing the same set of loads.

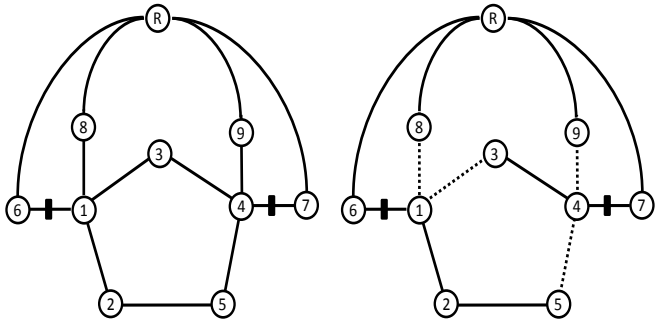
B. Island graph

TABLE I
Mapping of vertices and edges to construct island graph.

Island Graph		IEEE Test Feeder
Switch	Node	Load ID
w_1	$F_2, v_8 - v_1$	$(250 - 251)$
w_2	$v_1 - v_2$	$(18 - 135)$
w_3	$v_2 - v_3$	$(151 - 300)$
w_4	$v_4 - v_5$	$(97 - 197)$
w_5	$v_4 - F_3, v_7$	$(450 - 451)$
w_6	$v_3 - v_4$	$(54 - 94)$
w_7	$v_1 - v_3$	$(13 - 152)$
w_8	$F_1, v_6 - v_1$	$(149 - 150)$
w_9	$F_1, v_6 - F_4, v_9$	$(95 - 195)$

To simplify the analysis of the switch configuration problem we modify the island graph. This is done by the following steps. (1) Loads from the island Graph are converted to vertices in the graph. (2) Feeders $F_1 \ldots F_4$ are turned into vertices. (3) Switches in Island Graph are converted into undirected edge in the graph. (4) A virtual root node and directed edges $(v_r \rightarrow F_j)$ for all feeders are added.

The result is the Island graph in Figure 2(a). A complete mapping between the reduced feeder and the Island graph is given in Table I. For example, switch w_2 connects nodes v_1 and v_2 in the island Graph, or loads 151 and 300 in the test feeder. In the Island graph, it is now an edge between the vertices v_1 and v_2.



(a) (b)

Fig. 2. (a) Island graph showing potential topologies achievable. (b) Example spanning tree in network.

The Island graph $G = (V, E)$ is the network used in the remaining analysis. Denote the added edges $\tau = \{e \in E : e = (v_r, F_i)\ \forall F_i\}$. This construction leads to a simple method for enumerating each valid topology of the Island graph.

Consider $\mathcal{T}$ to be any spanning tree over G and $\mathbb{T}$ the set of all spanning trees constructed on G. We refer to the set of spanning trees containing subtree τ as $\mathbb{T}_\tau$. Figure 2 represents an example spanning tree that can be constructed. The number of spanning trees is indicated as $|\mathbb{T}| = T$.

Proposition 1: The set $\mathbb{T}_\tau$ represents all valid switch configurations.

Proof: Any $\mathcal{T} \in \mathbb{T}_\tau$ will maintain that every island is connected. Additionally no feeders are connected besides through the virtual node. ∎

Enumerating every spanning tree of the island graph which includes subtree τ will generate the set of topologies corresponding to valid switch configurations. Note the naive method of enumerating all $2^{|\mathbf{w}|}$ switch positions will enumerate all spanning forests of the graph which include the spanning trees associated with normal operation as well as the spanning tree under any number of outages where there are disconnected loads. In related work, the authors address the outage detection problem under the same measurement, detection and optimality criterion in [17].

C. Load Model

Each node v_n excluding the virtual root and feeder nodes in the graph has a consumption load x_n. We assume for now that the load is single time invariant value. The pseudo-measurement of each load is $\hat{x}_n$. We denote the pseudo-measurement error of load x_n by $\epsilon_n = x_n - \hat{x}_n$. We assume that the loads are single phase real power quantities and the errors are mutually independent random variables: $\epsilon_n \sim N(0, \sigma_n^2)$. Therefore, $x_n \sim N(\hat{x}_n, \sigma_n^2)$. $\mathbf{x}$ and $\hat{\mathbf{x}}$ represent the vector of true loads and that of load pseudo-measurements; thus $\mathbf{x} \sim N(\hat{\mathbf{x}}, \mathbf{\Sigma})$. Covariance matrix $\mathbf{\Sigma}$ being diagonal.

D. Measurement Model

For any edge e, we denote by s the power flow on it to all active downstream loads. The measured flow depends on unknown network topology and true loads.

The sensor placement $\mathcal{M} \subset E$, is a subset of edges of the network. We assume that the magnitude and direction of power flow is measured. Additionally, we assume that the power flow measurements are noise free, with errors only with load pseudo-measurements.

In general for a topology $\mathcal{T} \in \mathbb{T}$ and measurements $\mathbf{s} = \{s_1, \ldots, s_M\}$ the k^{th} flow is $s_k(\boldsymbol{x}) = \sum_{v_j \in V_k(\mathcal{T}, \mathcal{M})} x_j$. Where $V_k(\mathcal{T}, \mathcal{M})$ is the subset of downstream nodes for a particular placement and spanning tree.

Given a general graph where all nodes can have some consumptions, we can represent the observations with

$$\mathbf{s}(\mathcal{T}, \mathbf{x}) = \Gamma(\mathcal{T}, \mathcal{M})\mathbf{x} \tag{1}$$

where $\Gamma(\mathcal{T}, \mathcal{M}) \in \{0, 1\}^{|E| \times |V|}$.

For instance the sensor placement in Figure 2(a) (where all feeder nodes have 0 consumption) we have:

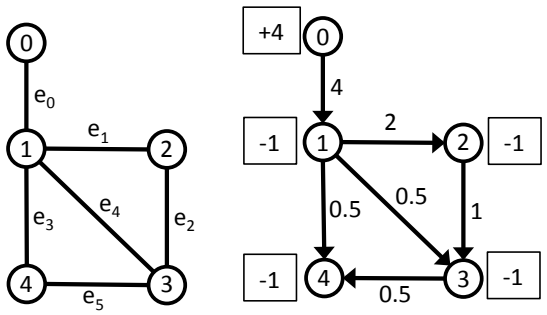
1) sensor placement $\mathcal{M} = \{e_1, e_9\}$.
2) subsets $V_1(\mathcal{T}) = \{v_1, v_2, v_5\}$ and $V_2(\mathcal{T}) = \{v_3, v_4\}$

The observation matrix is then the following:

$$\Gamma(\mathcal{T}, \mathcal{M}) = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{2}$$

E. Network Flow Representation

In much of this work, we use the combinatorial graph and network flow representations to solve various the detection and placement problems. Relaxing the spanning tree assumption for now, we can represent network flows on a graph as follows.



(a) (b)

Fig. 3. 3(a) Simple undirected graph. 3(b) Network flows and direction (on edge), consumption and production of value x_n (boxed) indicated.

Given a graph incidence matrix $B \in \{-1, 0, +1\}^{|V| \times |E|}$, where each undirected edge has a pre specified direction: $e_k = (v_n, v_m)$ on which to assign columns of B as follows.

$$B_{i,j} = \begin{cases} +1 & \text{if } v_i \text{ is the } \textit{originating node} \text{ of edge } e_j \\ -1 & \text{if } v_i \text{ is the } \textit{terminal node} \text{ of edge } e_j \\ 0 & \text{else} \end{cases} \tag{3}$$

We then denote the flows on each edge of the graph as $\mathbf{f} \in R^{|E|}$ and the consumption vector $\mathbf{y} = [1^T\mathbf{x}\ -\mathbf{x}]^T$. Therefore, the flow constraints can be represented as:

$$B\mathbf{f} = \mathbf{y}. \tag{4}$$

For example, the flow constraints for the undirected graph in Figure 3(a) is the following:

$$\begin{bmatrix} +1 & 0 & 0 & 0 & 0 & 0 \\ -1 & +1 & 0 & +1 & +1 & 0 \\ 0 & -1 & +1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & +1 \\ 0 & 0 & 0 & -1 & 0 & -1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}. \tag{5}$$

As an example, the following solution holds: $\mathbf{f} = [4\ 2\ 1\ 0.5\ 0.5\ 0.5]^T$ and $\mathbf{y} = [4\ -1\ -1\ -1\ -1]^T$ as shown in Figure 3(b). Note that negative f_i indicate a flow opposite to the arbitrary direction specified in the incidence matrix. The sign of y_i indicates a source (positive) or sink (negative) of energy. The graph indicates that v_0 is a single source for each sink in the network.

Assume a spanning tree $\mathcal{T}$ is encoded by the vector $\mathbf{w}$, indicating the edges that are removed. Then the incidence matrix of a particular tree $B_{\mathbf{w}} \in \mathbb{R}^{|V|\times|V|-1}$, will satisfy $B_{\mathbf{w}}\mathbf{f} = \mathbf{y}$. In the case of a spanning tree, the reduced matrix $B^r_{\mathbf{w}}$, where the first column is removed is an $|V|-1\times|V|-1$ invertible matrix. We can represent the matrix $\Gamma(\mathcal{T},\mathcal{M})$ by the following procedure.

$$\Gamma(\mathcal{T},\mathcal{M}) = A(\mathcal{M})B^{r,-1}_{\mathbf{w}} \tag{6}$$

Matrix $A(\mathcal{M})$ indicates the edges that are being measured, so $A(k,m_k) = 1, \forall m_k \in \mathcal{M}$. The superscript B^r indicates that the first row the matrix has been removed.

IV. Deterministic Case

A. Detector Operation

Given nodal consumptions $\mathbf{x}$ and observed flow $\mathbf{s}$ the following deterministic program is proposed:

$$\text{find } \mathbf{f}, \mathbf{w} \tag{OPT-1}$$

s.t.

$$|f_i| \leq |\mathbf{x}|w_i \tag{7}$$

$$B\mathbf{f} = \mathbf{y} \tag{8}$$

$$f_i = s_i \ \forall e_i \in E(\mathcal{M}) \tag{9}$$

$$\mathbf{w} \in \text{ SP-TREE} \tag{10}$$

The feasibility program is mixed integer in with boolean vector $\mathbf{w} \in \{0,1\}^{|E|}$ for the edges in G as well as the flow along each edge $\mathbf{f} \in R^{|E|}$. The network flow constraint is given in (7). Constraint (7), limits the edge flows to either be set to zero, or be fully unconstrained. Constraint (9), sets each observed edge to the sensor value while (10) constrains the status of edges in G to form a spanning tree.

$$\hat{\mathcal{T}} = \{\mathcal{T} \in \mathbb{T} : \mathbf{s} = \mathbf{s}(\mathcal{T},\mathbf{x})\} \tag{11}$$

The naive detector in (11) will enumerate every spanning tree, then given the nodal consumptions, evaluate the theoretical flow value $s(\mathcal{T})$ and compare it to the observed flow. There are two questions regarding this simple technique:

1) *Correctness*: How to guarantee that this method will return a unique and correct spanning tree. How to guarantee that each spanning tree results in a unique flow observation.
2) *Efficiency*: Enumerating each spanning tree then performing a comparison will lead to a detector of $O(|\mathbb{T}|)$ complexity. Typically $|\mathbb{T}|$ grows exponentially in the size of the graph.

The question of correctness is solved in IV-B and an efficient polynomial complexity algorithm is introduced in IV-D.

B. Detector Objective: Spanning Tree Identifiability

We only require a sensor placement such that different spanning trees lead to unique measured flow values. The following definition is of use:

Definition 1: The set $\mathbb{T}$ is identifiable if $\forall \mathcal{T}, \mathcal{T}' \in \mathbb{T}$ where $\mathcal{T} \neq \mathcal{T}'$ we have that $\mathbf{s}(\mathcal{T},\mathbf{x}) \neq \mathbf{s}(\mathcal{T}',\mathbf{x})$.

In the deterministic case, we desire a placement $\mathcal{M}$ such that $\mathbb{T}$ is identifiable. A naive method of determining whether $\mathbb{T}$ is identifiable is to enumerate every single spanning tree, generate $\Gamma(\mathcal{T},\mathcal{M})$ to verify that for no two trees $\Gamma(\mathcal{T},\mathcal{M}) = \Gamma(\mathcal{T}',\mathcal{M})$. This naive procedure has $O(|\mathbb{T}|^2)$ complexity, which can be quite intensive since $|\mathbb{T}|$ is generally very large.

C. Deterministic Sensor Placement

An important result is the following:

Theorem 1: $\mathbb{T}$ is identifiable if and only if the graph $E(G)\setminus E(\mathcal{M})$ of the Island graph forms a spanning tree.

We refer to any placement $\mathcal{M}$ satisfying Theorem 1 as a *valid placement*.

The proof relies on graph theoretic arguments as opposed to linear algebraic arguments on the relation in eq. (6). The intuition of Theorem 1 is that to have observability of all spanning trees, we must have a sensor placement such that any cycle that can be constructed on G will have some flow sensor on it. This corresponds to the dimension of the cycle space, referred to as the circuit rank $\mu(G) = |E| - |V| + 1$, which is the minimum number of measurements needed to correctly detect all spanning trees on G. This gives us a $O(E)$ verifiable condition to ensure that all spanning trees are identifiable as opposed to $O(|\mathbb{T}|^2)$ with the naive method since we only need to remove particular edges of the graph and test if it is a spanning tree.

Theorem 1 provides a way to construct the set of all placements where identifiability is achieved.

First consider the function $h(\mathcal{T}) = E(G) \setminus E(\mathcal{T})$, which merely extracts the edges in G that are not in $\mathcal{T}$. An obvious consequence to Theorem 1 is the following:

Corollary 1: The function $h : \mathcal{T} \to \mathcal{M}$ is a bijection between the set $\mathbb{T}$ and the set of all valid placements $\mathbb{M}$.

The following is useful:

Remark 1: Corollary 1 implies that $|\mathbb{M}| = |\mathbb{T}|$.

Corollary 1 is quite important from a placement perspective since it actually yields a method to generate a valid placement in the deterministic case. Also, it allows us to enumerate *all* valid placements for a graph. This is important when dealing with a stochastic case where sensor placement relies on mostly evaluating each placement in $\mathbb{M}$.

In the case of valid placements on the Island graph this set is restricted, since having sensors on edges in τ, since they have no physical meaning. The restricted set is given by $\mathbb{M}_\tau = \{f(\mathcal{T}) | \mathcal{T} \in \mathbb{T}_\tau\}$.

D. Spanning Tree Detection via Relaxed Flow Solution

Theorem 1 states that if $E(G)\setminus E(M)$ forms a spanning tree, there is a unique observable flow for each unique spanning tree. However, it provides no efficient method of actually decoding the correct spanning tree.

$$\mathbf{f}^\star = \{\mathbf{f} : \text{ st. } B\ \mathbf{f} = \mathbf{y} \text{ and } f_i = s_i \ \forall e_i \in E(\mathcal{M})\}. \tag{12}$$

We show that solving a combinatorial search over all spanning trees, reduces to solving a relaxed form of the problem, over the fully connected graph. Surprisingly, the solution to the linear equation over $\mathbf{f} \in R^{|E|}$ recovers the sparsity pattern

in $\mathbf{f}$ corresponding to a spanning tree without any sparsity inducing heuristics.

We can represent the conservation of flow in eq. (4), by partitioning the incidence matrix B and flow vector f into observed $(B_M, \mathbf{f}_M)$ and non-observed $(B_N, \mathbf{f}_N)$ components.

Therefore, we can represent the network flow with the following:

$$\begin{bmatrix} B_N^r & B_M^r \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathbf{f}_N \\ \mathbf{f}_M \end{bmatrix} = \begin{bmatrix} \mathbf{x} \\ \mathbf{s} \end{bmatrix} \tag{13}$$

Lemma 1: For the sensor placement condition in Theorem 1 where $E(G) \setminus E(\mathcal{M})$ is a spanning tree, $\text{rank}(B_N) = N-1$.

The matrix $B_N \in R^{|V|\times|N-1|}$ is of rank $N-1$. The sub matrix B_N^r with the first row removed is invertible, which leads to the following solution to (12).

$$\mathbf{f}(\mathbf{x}, \mathbf{s}) = (B_N^r)^{-1}(\mathbf{x} - B_M^r \mathbf{s}) \tag{14}$$

Where we have removed the first row of B_N and B_M. Next we must show that the solution to this is in fact the correct spanning tree on the graph.

Theorem 2: If $\mathcal{M}$ satisfies the condition in Theorem 1, the solution vector $\mathbf{f} = [\mathbf{f}(\mathbf{x}, \mathbf{s})\ \mathbf{s}]^T$ encodes spanning tree $\mathcal{T}$.
The search over the set of spanning trees, can be replaced by solving a set of linear equations eq (12).

V. Stochastic Case: Detection

A. Maximum a Posteriori Spanning Tree Detector

Recall that in the stochastic case we integrate (1) error free flow information on edges (2) load pseudo measurements at nodes. Combining these two leads to a hypothesis testing problem as follows: Given the forecasted loads $\hat{\mathbf{x}}$, the true loads at each node are given as: $\mathbf{x} \sim N(\hat{\mathbf{x}}, \sigma^2 I)$. Therefore, under a particular hypothesized spanning tree $\mathcal{T}$, the true flow would be distributed as:

$$\begin{aligned} \mathbf{s} \mid \{\mathcal{T}_i,\ \hat{\mathbf{x}}\} &\sim N(\Gamma_i \hat{\mathbf{x}},\ \sigma^2 \Gamma_i \Gamma_i^T) && (15)\\ &\sim N(\mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}}),\ \Sigma_i) && (16)\\ &= \mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}}) + \epsilon_{s,i} && (17) \end{aligned}$$

Each line is an equivalent definition. The matrix Γ_i in eq. 15, denotes $\Gamma(\mathcal{T}, \mathcal{M})$ for tree $\mathcal{T}_i$ under a known, fixed placement The term $\mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}})$ in eq. 16, 17 indicates the theoretical flow that should be observed under the forecast of nodal consumption. The error $\epsilon_{s,i}$ in eq. 17 is a zero mean multivariate gaussian with covariance matrix Σ_i. Assuming uniform prior on hypotheses, the general MAP detector reduces to a maximum likelihood detection of each spanning tree:

$$\mathcal{T} = \arg\max_{\mathcal{T}_i \in \mathbb{T}} \Pr\left(\mathbf{s} \mid \hat{\mathbf{x}},\ \mathcal{T}_i\right) \tag{18}$$

The MAP detector is optimal in minimizing the mean square error but suffers from the $O(|\mathbb{T}|)$ complexity of the deterministic case.

B. Flow Based Approximate MAP Detector

The flow solution in eq. (13) can be used to construct an efficient hypothesis detector which does not have $O(|\mathbb{T}|)$ complexity. Recall in the deterministic case, the flow is $\mathbf{f}(\mathbf{x}, \mathbf{s})$. In the stochastic case, $\mathbf{s}$ and $\hat{\mathbf{x}}$ is given, therefore computing $\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s})$, can be useful. The goal is to compute the flow given the measurements available ($\mathbf{s}$ and $\hat{\mathbf{x}}$), and use it for detection of the correct spanning tree. First, we can determine the distribution of this flow vector conditioning of a candidate hypothesis $\mathcal{T}_i$:

$$\begin{aligned} \mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}) \mid \{\mathcal{T}_i\} &= B_N^{r,-1}(\hat{\mathbf{x}} - B_M^r \mathbf{s}(\mathcal{T}_i, \mathbf{x})) && (19)\\ &= B_N^{r,-1}(\hat{\mathbf{x}} - B_M^r(\mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}}) + \epsilon_{s,i})) && (20)\\ &= \mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}})) + B_N^{r,-1} B_M^r \epsilon_{s,i} && (21)\\ &\sim N(\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}})), \Sigma_{f,i}) : \mathcal{T}_i \text{ is true} && (22) \end{aligned}$$

The LHS of eq. (19) is the distribution of the 'noisy-flow' conditioning on a particular hypothesis $\mathcal{T}_i$. The RHS evaluates the flow equation (14) using the forecasted consumption $\hat{\mathbf{x}}$ instead of the true value $\mathbf{x}$. Finally, in (21)-(22) the relationship between $\mathbf{s}(\mathcal{T}_i, \mathbf{x})$ and $\mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}})$, leads to eq. (15)-(17).

The vector $\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}}))$ is the flow from spanning tree $\mathcal{T}_i$ and nodal injections $\hat{\mathbf{x}}$. The true 'noisy-flow' is distributed around this value, depending on the hypothesis $\mathcal{T}_i$.

A possible hypothesis test is the following:

$$\hat{\mathcal{T}} = \arg\max_{\forall \mathcal{T}_i \in \mathbb{T}}\ \Pr(\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}) \mid \mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}}))). \tag{23}$$

This is no better than (18) for the following reasons:

1) We must compute the hypothesis mean $\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}(\mathcal{T}_i, \hat{\mathbf{x}}))$ under every spanning tree $\mathcal{T}_i$, it is still of $O(|\mathbb{T}|)$ complexity.
2) The covariance matrix is of rank $\mu(G)$ (the rank of Γ_i) and not E (the size of $\mathbf{f}$). Therefore $\Sigma_{f,i}^{-1}$ is not positive definite and therefore the inverse $\Sigma_{f,i}^{-1}$ cannot be computed.

An alternative is to test $\mu(G)$ elements of the 'noisy-flow' $\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s})$ under the hypothesis that their true value is zero corresponding to the zero's of the hypothesized spanning tree.

Using the following shorthand: $\mathbf{f}_o = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{s})$ and $\mathbf{f}_\mathcal{T} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{s}(\mathcal{T}, \hat{\mathbf{x}}))$ we can represent the variables in (23). Under hypothesis $\mathcal{T}$, $\mathbf{f}_\mathcal{T}(i) = 0$ for some set of indices $i \in \mathcal{I}_\mathcal{T}$ where $\mathcal{I}_\mathcal{T} = \{i : e_i \in E(G) \setminus E(\mathcal{T})\}$.

Therefore, under a hypothesis $\mathcal{T}$, we can calculate the following likelihood:

$$\hat{\mathcal{T}} = \arg\max_{\mathcal{I}_\mathcal{T}: \mathcal{T} \in \mathbb{T}} \Pr(\mathbf{f}_o(i_1), \ldots, \mathbf{f}_o(i_\mu) | \mathbf{f}_\mathcal{T}(i_1) = 0, \ldots, \mathbf{f}_\mathcal{T}(i_\mu) = 0). \tag{24}$$

In this case, the reduced covariance matrix is potentially invertible. The following theorem relates the combinatorial test to a test of zero flows on the empirical flow.

Theorem 3: The zero flow hypothesis detection in (24) and the combinatorial flow hypothesis test in eq. (18) are equivalent for $\mathcal{M}$ satisfying the placement condition in Theorem 1 of size $\mu(G)$.

Therefore the MAP detector over the set of all spanning trees reduces to testing whether an observation vector Unfortunately we still need to enumerate $|\mathbb{T}|$ hypotheses.

Fortunately, however vector form allows us to very efficiently prune out all but a few alternative hypothesis to test.

Intuitively, $\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s})$ will have some very few small values which actually encode potential spanning trees, and many very large values which can just be pruned.

This leads to the following two step approximate MAP detector:

1) Evaluate the empirical flow $\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s})$ via (19).
2) Compute the minimum weight spanning tree solution on graph G where edges are weighted with $-|\mathbf{f}(\hat{\mathbf{x}}, \mathbf{s})|$.

C. Cycle Descent Approximate MAP Detection

An alternative approximate MAP detector is based on generating single cycle edge exchanges $\Delta e_i = \{e_i \rightarrow e_i'\}$. For ever edge in e_i in the co-tree of current tree, the perturbed edge is chosen to be an edge along the fundamental cycle of e_i. The complete algorithm is given as the following: The

Input: [1] Observed flows $\mathbf{s}$.
[2] Load Forecast $\hat{\mathbf{x}}$ and Error Covariance Σ
[3] Graph G
Output: MAP Detection Hypothesis $\mathcal{T}$

```
// Must start with feasible Tree
T ← feasible-tree(s_o, x̂, G)
FC ← generate − fundamental − cycle (T)
while Δloglik ≠ 0 do
    for c_k ∈ FC do
        loglik, T ← local − update(T, c_k, s, x̂)
        FC ← update − cycles (T, FC)
    end
end
```

cycle descent algorithm performs as follows:

1) **feasible-tree** *:* A feasible tree is generated which satisfies the pattern of observed zeros as the observed flows $\mathbf{s}$. The edges of G are given weights as follows:

- Edges with measurements are weighted as follows: zero measurements have weight 0.
- Non-zero measurements are weighted with some positive value K.
- Remaining edges are assigned a very large weight ($\approx |E|\ K$).

Finally, a maximum weight spanning tree is calculated on the weighted graph. This ensures that the initial spanning tree $\mathcal{T}_0$ satisfies $\mathbb{I}\{\mathbf{s} > 0\} = \mathbb{I}\{\mathbf{s}(\mathcal{T}, \hat{\mathbf{x}}) > 0\}$. The remaining spanning tree is feasible, where evaluating $\Pr(\mathbf{s} \mid \mathcal{T}_i,\ \hat{\mathbf{x}})$ will lead to a finite (albeit small value).

One would assume that the maximum spanning tree output of FMST method can be used as a suitable initial tree. Unfortunately, the output of the FMST is not guaranteed to produce a tree that satisfies the observed non-zero flow. Edges that measured zero flow will not be included in the initial condition tree, however we are not guaranteed to include all edges with non-zero measured flow.

2) **local-update** *:* For cycle $\mathbf{c}_k$ and edge e_k, the next edge is chosen by evaluating the likelihood of function $\mathcal{T} - e_k + e_j$, where $e_j \in E(c_k)$. Not every The next edge e_j is chosen which maximizes the likelihood.

3) **update-cycles** *:* After each edge exchange operation, the cycles must be updated. A queue is maintained for the edge e_k to be processed, where the elements are updated.

D. Local MAP Search

A simple local search procedure is the following:

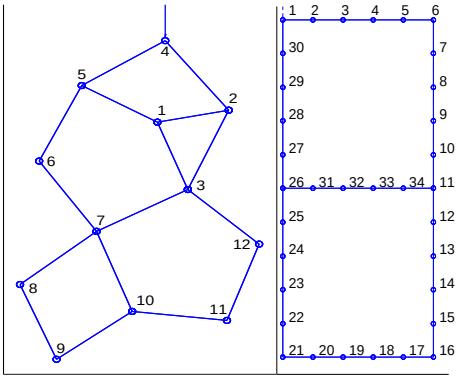
1) An approximate detection is performed: (using FMST or Cycle Descent) with output $\mathcal{T}^\star$.
2) A subset of spanning trees 'close' to $\mathcal{T}^\star$ are enumerated with likelihood function evaluated.

We define trees near $\mathcal{T}^\star$, to be all spanning trees that map to the same fundamental cycle basis. Enumerating the candidate edges of this set is of complexity $O(|E|\mu(G))$ and is discussed in [18].

VI. Numerical Experiment

A. Deterministic Placement

We test the placement problem on a set of planar graphs, shown in Figure 2. In both graphs, designate a single vertex as the source, which is indicated in the dashed horizontal lines. Graph G_1 has $v_{\text{root}} = v_4$ and G_2 has $v_{\text{root}} = v_1$.



(a) (b)

Fig. 4. Sample graphs used to benchmark deterministic and stochastic cases, G_1 with $v_{\text{root}} = v_4$ ((Figure 4(a)). These graphs have high circuit rank but smaller $|E|/\mu(G)$, therefore difficult to visualize stochastic placement. Graph G_2 (Figure 4(b)) used to illustrate stochastic placement as well as provide sub modularity counterexample.

TABLE II
Deterministic Topology Detection.

	$\mu(G)$	$\lvert\mathbb{T}\rvert$	ϵ'			$\lvert E\rvert/\mu(G)$
			mean	std.	max/min	
G_1	5	391	56.9	26.7	299/6	2.8
G_2	5	830	139.3	72.2	185/10	3.2

To test the placement problem, we enumerate the set of spanning trees for each of the graphs. The method relies on the backtracking method developed in [19]. The simulation was implemented in MATLAB and deemed correct by checking that each spanning tree was unique and the number of test trees corresponded to those calculated from the matrix-tree

theorem [20], where $T = \det(L_v)$. Where L_v is the v minor of the laplacian matrix with the result being invariant to v.

For the three graphs in Figure 4, the graph statistics and experiment results are shown in Table II. We evaluate the experimental error rate

$$\epsilon = \frac{1}{N(N-1)} \sum_{i \neq j} \mathbb{I}\{\mathbf{s}(\hat{\mathcal{T}}_i, \mathbf{x}) \neq \mathbf{s}(\mathcal{T}_i, \mathbf{x})\}. \quad (25)$$

From Theorem 1, the missed detection error must be zero. The computed ϵ was zero in both cases, as was expected.

We evaluate the output according to an arbitrary input because we would like to compare it to the case where only magnitude and not direction is measured. This is a common type of power system measurement, where no phase is reported and power direction flow is unknown. In this case, we evaluate ϵ' which now compares $|\mathbf{s}(\hat{\mathcal{T}}_i, \mathbf{x})| \neq |\mathbf{s}(\mathcal{T}_i, \mathbf{x})|$ instead. The computed values for ϵ' are shown in Table II. We evaluate each valid placement in $\mathbb{M}$ to illustrate the importance of flow direction. The value reported in Table II is the mean missed detection $\pm$ the standard deviation. This verifies that different placements result in different unsigned missed detection rates. We see that if the direction of flow is not known, around 10% of the spanning trees are indistinguishable on average.

B. MAP Detection Performance

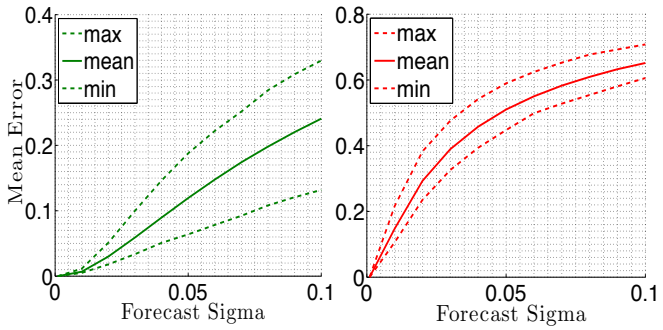


Fig. 5. Mean missed detection error for the graphs G1 (5(a)), G2 (5(b) with respect to σ.

The performance of the MAP detector is evaluated for each of the Graphs in Figure 4. The one shot detector performance is evaluated with a uniform load mean of $\mu_i = 1$ and forecast error of σ. Figure 6 shows the mean missed detection error over all hypotheses with respect to σ.

A number of important observations can be seen from this analysis. Different graphs experience widely different behavior. For example G_1 has many very short cycles where many of the spanning trees will see multiple zeros while G_2 has only two cycles with high edge count per cycle. The mean errors in G_1 are must smaller (roughly half) as that of G_2.

The sensor placement in the stochastic case has a dramatic impact on the mean missed detection error. This is slightly counter-intuiive, since a single placement which maps to a single spanning tree must correctly decode all spanning trees with low error. A symmetry between placement and tree's would make one naively guess that the missed detection error should not depend on any single placement. Within a graph, it is observed that the the mean length of all fundamental cycles associated with a placement is slightly correlated with the mean error.

C. Approximate MAP Detector

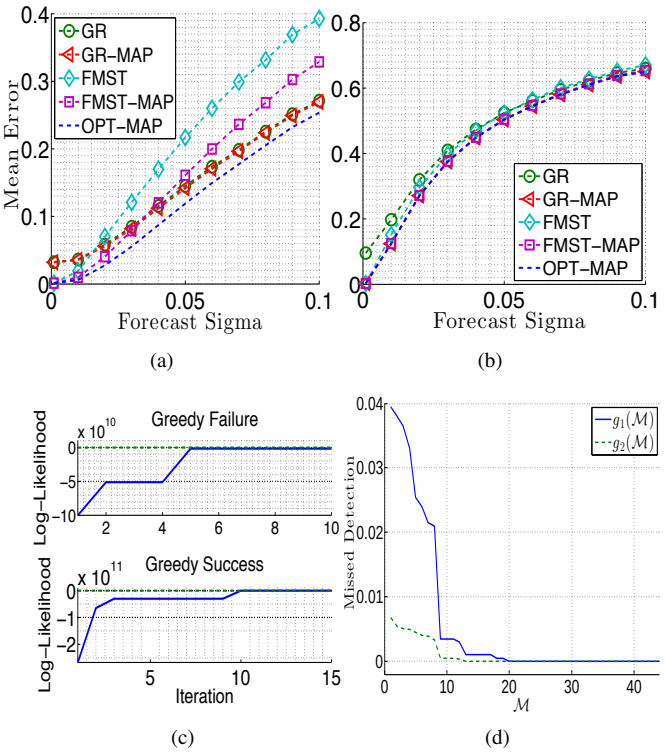


Fig. 6. Comparison of mean missed detection error for optimal MAP detector, flow based approximate detector (FMST/FMST-MAP) and cycle descent detector (GR/GR-MAP) for the graphs G1 (5(a)), G2 (5(b)) with respect to σ. (6(c)) Example of cycle descent success and failure for two spanning trees. (6(d)) Evaluated missed detection probabilities for the set of placements on the network $\mathbb{M}_\tau$. For half of the placements, the conditional pdfs are far enough that there is negligible error.

The two approximate MAP algorithms are tested on G_1 and G_2, where the performance are shown in Figure 6(a)-6(b) respectively.

1) Flow-Maximum Weighted Spanning Tree (FMST): In the near noiseless case, the flow based approximate MAP detector performs identically to the combinatorial map detector, in both G_1 and G_2. This is because the 'noisy-flow' values are very close to their correct values of zero. For this reason, the maximum weighted spanning tree algorithm can correctly reconstruct the correct spanning tree.

In the high noise case, the algorithm fails worst in the high cycle count graph G_1, where the maximum spanning tree graphs very rarely match with the maximum likelihood output. In G_2, however the two are nearly identical since the tree does not involve many Subsequent local search leads to an improvement in the mean missed detection rate.

2) Cycle Descent Algorithm: The cycle descent algorithm has a very different performance than the FSMT algorithm. For both graphs, the performance is similar, in that a small

subset of trees in G_1 will always fail regardless of SNR. For the remaining spanning trees, the algorithm converges to the MAP output. In the case of G_1, only 4% of the trees lead to a failure of the algorithm, for the used placement. Different sensor placements lead to different values of non-convergence.

D. 123 Test Feeder

As presented in Section I, we introduced a method of transforming the IEEE 123 test feeder into a reduced feeder (Figure 1(b)) and an island graph (Figure 2(a)). The results we present allow us some insight into the topology detection problem. From Theorem 1, the minimum number of flow measurements is $\mu(G) = 4$. Since the actual set of allowable trees is $\mathbb{T}_\tau$, this restricts the set of trees to search over, in evaluating for each sensor placement:

1) Mean missed detection error over all possible spanning trees:
$$g_1(\mathcal{M}) = \sum_{\mathcal{T}\in\mathbb{T}} \Pr(\mathcal{T}) \Pr(\hat{\mathcal{T}} \neq \mathcal{T}|\mathcal{T};\mathcal{M}). \tag{26}$$
2) Maximum missed detection error over all possible spanning trees:
$$g_2(\mathcal{M}) = \max_{\mathcal{T}\in\mathbb{T}} \Pr(\hat{\mathcal{T}} \neq \mathcal{T}|\mathcal{T};\mathcal{M}). \tag{27}$$

For simplicity assume a single phase flow of only real power. Given the three phase complex power consumption of each load, we reduce this to a single value power value. This is the case, if the measurement on a line sum and report the sum of the three phases. For a rough estimate of the forecast error, we follow the forecasting model in [21], where day ahead forecasts are computed for varying aggregation levels. The forecast errors are used to construct the following scaling law for coefficient of variation:

$$\frac{\sigma}{\mu} = \sqrt{\frac{3562}{W} + 41.9}. \tag{28}$$

. Since the loads of each individual island is quite large and beyond the critical load reported in [21], the CV of each island is close to the irreducible error of 6.3%. Figure 6(d) shows the performance of an initial stochastic placement for the IEEE123 test feeder. We evaluate the set of restricted placements $\mathbb{M}_\tau$ and spanning trees $\mathbb{T}_\tau$, where $|\mathbb{M}_\tau| = 44$. Notice that for almost half of the placement the maximum error is negligibly small. One way to consider this is that although the various hypotheses means have signification error variances, some placements separate out the hypotheses sufficient such that they are all distinguishable.

This example allows us to comment on the common understanding of generalized state estimation. In current practice, determining the boolean values of breakers or In practice the set of all topologies generally map to range of $\pm 5\%$. Therefore, when factoring in uncertainties, the missed detection rates can be high. On the other hand, measuring flows lead to very large changes in the observation vector over the range of hypothesis.

VII. Conclusion

We formulate the problem of detecting switch configurations on residential feeders as a spanning tree detection problem on an 'Island graph'. The detection problem relies on power flow measurements on edges as well as load information at nodes. The deterministic case leads to notion of tree identifiability and sensor placement conditions to ensure identifiability. For the stochastic case, we propose a greedy algorithm which leads to a locally optimal solution.

References

[1] J. Lavaei, D. Tse, and B. Zhang, "Geometry of power flows in tree networks," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–8.

[2] A. Lam, B. Zhang, and D. Tse, "Distributed algorithms for optimal power flow problem," *arXiv preprint arXiv:1109.5229*, 2011.

[3] M. Farivar, R. Neal, C. Clarke, and S. Low, "Optimal inverter var control in distribution systems with high pv penetration," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–7.

[4] A. Lam, A. Dominguez-Garcia, B. Zhang, and D. Tse, "Optimal distributed voltage regulation in power distribution networks," Tech. Rep., 2012.

[5] P. Jahangiri and D. C. Aliprantis, "Distributed volt/var control by pv inverters," *Power Systems, IEEE Transactions on*, vol. 28, no. 3, pp. 3429–3439, 2013.

[6] J. Smith, W. Sunderman, R. Dugan, and B. Seal, "Smart inverter volt/var control functions for high penetration of pv on distribution systems," in *Power Systems Conference and Exposition (PSCE), 2011 IEEE/PES*. IEEE, 2011, pp. 1–6.

[7] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.

[8] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC Press, 2004.

[9] G. N. Korres and P. J. Katsikas, "Identification of circuit breaker statuses in wls state estimator," *Power Systems, IEEE Transactions on*, vol. 17, no. 3, pp. 818–825, 2002.

[10] G. N. Korres and N. M. Manousakis, "A state estimation algorithm for monitoring topology changes in distribution systems," in *Power and Energy Society General Meeting, 2012 IEEE*. IEEE, 2012, pp. 1–8.

[11] R. Arghandeh, M. Gahr, A. von Meier, G. Cavraro, M. Ruh, and G. Andersson, "Topology detection in microgrids with micro-synchrophasors," *arXiv preprint arXiv:1502.06938*, 2015.

[12] G. Cavraro, R. Arghandeh, G. Barchi, and A. von Meier, "Distribution network topology detection with time-series measurements," in *Innovative Smart Grid Technologies Conference (ISGT), 2015 IEEE Power & Energy Society*. IEEE, 2015, pp. 1–5.

[13] D. Deka, S. Backhaus, and M. Chertkov, "Structure learning in power distribution networks: Part i," *arXiv preprint arXiv:1502.07820*, 2015.

[14] ——, "Structure learning in power distribution networks: Part ii," *arXiv preprint arXiv:1502.07820*, 2015.

[15] S. Bolognani, N. Bof, D. Michelotti, R. Muraro, and L. Schenato, "Identification of power distribution network topology via voltage correlation analysis," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 2013, pp. 1659–1664.

[16] Y. Sharon, A. M. Annaswamy, A. L. Motto, and A. Chakraborty, "Topology identification in distribution network with limited measurements," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*. IEEE, 2012, pp. 1–6.

[17] R. Sevlian, Y. Zhao, R. Rajagopal, A. Goldsmith, and H. V. Poor, "Outage detection in power distribution networks using optimally deployed power measurements," *https://arxiv.org/abs/1111.11111*, 2008.

[18] M. M. Syslo, "On the fundamental cycle set graph," *Circuits and Systems, IEEE Transactions on*, vol. 29, no. 3, pp. 136–138, 1982.

[19] H. N. Gabow and E. W. Myers, "Finding all spanning trees of directed and undirected graphs," *SIAM Journal on Computing*, vol. 7, no. 3, pp. 280–287, 1978.

[20] R. Diestel, *Graph Theory {Graduate Texts in Mathematics; 173}*. Springer-Verlag Berlin and Heidelberg GmbH & Company KG, 2000.

[21] R. Sevlian, S. Patel, and R. Rajagopal, "Distribution system load and forecast model," *arXiv preprint arXiv:1407.3322*, 2014.

[22] R. B. Bapat, *Graphs and matrices*. Springer, 2010.

APPENDIX

$G = (V, E)$	Undirected Graph G; Vertices V; Edges E.
$\mathcal{T}$	Spanning tree on G.
$\mathbb{T}$	Set of all spanning trees that are constructed on G.
τ	Set of edges in constructing Island graph.
$\mathcal{M}$	Sensor placement ($\mathcal{M} \subset E$).
$\mathbb{M}$	Set of all sensor placements leading to identifiably in $\mathbb{T}$.
$x_n, \hat{x}_n$	True and forecasted load of node v_n.
σ_n^2, Σ	Forecast error variance and covariance matrix.
$\Gamma(\mathcal{T}, \mathcal{M})$	Observation matrix for tree $\mathcal{T}$, sensor placement $\mathcal{M}$.
$\mathbf{s}_{obs}$	Set of measured power flow.
$\mathbf{s}(\mathcal{T})$, $\hat{\mathbf{s}}(\mathcal{T})$	True and predicted flow measured under hypothesis $\mathcal{T}$.
r_{ij}	Acceptance region of $\mathcal{T}_i$ over $\mathcal{T}_j$.
R_i	Acceptance region of hypothesis h_i over all alternatives.
$g_{1,2}(\mathcal{M})$	Shorthand notation for (1) maximum / (2) mean missed detection probability over all hypotheses.
c	Cycle in graph G.
$\mathcal{C}(G)$	Cycle space of G or set of all possible cycles.
$\mathcal{FC}(\mathcal{T})$	Fundamental Cycle Basis of G constructed by $\mathcal{T}$.
$\mathcal{FC}_{\mathcal{M}}$	Fundamental Cycle Basis constructed by sensor placement $\mathcal{M}$.
λ_k	λ_k is k^{th} cycle in $\mathcal{FC}_{\mathcal{M}}$.
$\mu(G)$	Circuit Rank of Graph.
$n(G)$	Number of connected components.
ΔE	Edge exchange operation to generate new tree: $\mathcal{T} \to \mathcal{T}'$.
$\mathcal{K}(c)$	Cycle-Sensor Map indicating all sensors on cycle c.

A. Useful Graph Theory Definitions and Results

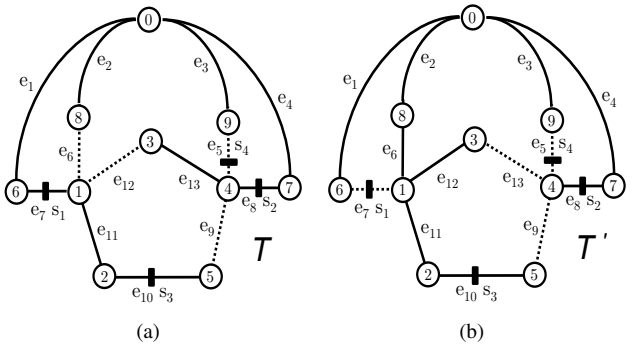


Fig. 7. Two spanning trees $\mathcal{T}$ (7(a)) and $\mathcal{T}'$ (7(b)), with sensor placement $\mathcal{M}$.

Refer to [20] for a more thorough presentation.

Cycle: A cycle $c = \{e_1, \ldots, e_N\}$ is a connected subgraph where each vertex is of degree 2.

Cycle Space: The set $\mathcal{B}_E$ is the power set over the edge set $\mathcal{B}_E = \{0, 1\}^{|E|}$. Any cycle c is a vector in the space $\mathcal{B}_E$. Vector addition is defined as $c' = c_1 \oplus c_2$ where new cycles are constructed via symmetric difference operation on edges: $E_1 \bigoplus E_2 = (E_1 \cup E_2) \setminus (E_1 \cap E_2)$. The cycle space $\mathcal{C}(G)$ of the graph is the vector space of all possible cycles in a particular graph.

For example consider the cycles in the graph G in Figure 7(a) (assume dashed lines now solid):

$$c_1 = \{e_7, e_1, e_2, e_6\}$$
$$c_2 = \{e_6, e_2, e_3, e_5, e_{13}, e_{12}\}$$
$$c_3 = c_1 \oplus c_2 = \{e_7, e_1, e_3, e_4, e_{13}, e_{12}\}$$

It is easy to see that c_1, c_2 and c_3 are all cycles in G.

Circuit Rank: The circuit rank of a graph is given by $\mu(G) = |E| - |V| + n(G)$ where $n(G)$ is the number of connected components of the graph. For example the Island graph in Figure 2(a) has $n(G) = 1$, and $\mu(G) = |13| - |10| + 1 = 4$.

Cycle Basis: The analog of a vector basis for cycle spaces is the cycle basis. A basis $\mathcal{B}_C \subset \mathcal{B}_E$ is the smallest number of cycles whereby all other cycles can be constructed via symmetric difference operations. The dimension of $\mathcal{C}(G)$ and $\mathcal{B}_C$ is $\mu(G)$, the circuit rank of the graph. Therefore, $\mu(G)$ is the smallest number of cycles required to produce all other cycles on a graph. We use $\mu(G)$ and μ interchangeably, whenever convenient.

For example, all cycles in the graph in Figure 2(a) can be constructed from $\mu(G)$ independent cycles which form the cycle basis of the graph.

Fundamental Cycle Basis: A Fundamental Cycle Basis $\mathcal{FC}$ is a cycle basis constructed using the following procedure: Given a spanning tree $\mathcal{T}$, enumerate the set of edges in G but not $\mathcal{T}$. Then for each $e \in E(G) \setminus E(\mathcal{T})$, construct $\mathcal{T} + e$ then find the single cycle c associated with e. So we can generate $\mu(G)$ cycles in $\mathcal{FC}$, which is the dimensionality of the basis. An equivalent definition for a Fundamental Cycle Basis is that each cycle will have one unique edge which is in no other cycle.

TABLE III
FUNDAMENTAL CYCLE BASIS FROM $\mathcal{T}$, $\mathcal{T}'$ AND CYCLE-MEASUREMENT MAP $\mathcal{K}(c)$

$\mathcal{FC}(\mathcal{T})$	$\mathcal{K}(c)$
$c_1 = \{e_6, e_2, e_1, e_7\}$	$\mathcal{K}(c_1) = \{s_1\}$
$c_2 = \{e_{12}, e_{13}, e_8, e_4, e_1, e_7\}$	$\mathcal{K}(c_2) = \{s_1, s_2\}$
$c_3 = \{e_5, e_3, e_4, e_8\}$	$\mathcal{K}(c_3) = \{s_2, s_4\}$
$c_4 = \{e_9, e_8, e_4, e_1, e_7, e_{11}, e_{10}\}$	$\mathcal{K}(c_4) = \{s_1, s_2, s_4\}$
$\mathcal{FC}(\mathcal{T}')$	$\mathcal{K}(c)$
$c_1 = \{e_7, e_6, e_2, e_1\}$	$\mathcal{K}(c_1) = \{s_1\}$
$c_2 = \{e_{13}, e_8, e_4, e_2, e_6, e_{12}\}$	$\mathcal{K}(c_2) = \{s_1, s_2\}$
$c_3 = \{e_5, e_8, e_4, e_3\}$	$\mathcal{K}(c_3) = \{s_2, s_4\}$
$c_4 = \{e_9, e_8, e_4, e_2, e_6, e_{11}, e_{10}\}$	$\mathcal{K}(c_4) = \{s_1, s_2, s_4\}$

From spanning trees $\mathcal{T}$ and $\mathcal{T}'$ in Figure 7(b), 7(b) we construct the following Fundamental Cycle Basis in Table III. Note that $\mathcal{FC}(\mathcal{T}) \neq \mathcal{FC}(\mathcal{T}')$. However, this does not always occur, see [18] for more details.

B. Proof of Theorem I

Recall, that theorem states that as long as $E(G) \setminus E(\mathcal{M})$ forms a spanning tree, then for any two trees $\mathcal{T}$, $\mathcal{T}'$, $s \neq s'$.

We now prove Theorem 1 in the following steps.

1 Introduce a cycle-measurement mapping object $\mathcal{K}(c)$ which tracks sensors on a cycle and show that an independence property if $E(G) \setminus E(\mathcal{M})$ forms a spanning tree.
2 We construct an edge exchange procedure which encodes the transition: $\mathcal{T} \to \mathcal{T}$ between any two spanning trees as a set of single cycle edge exchanges on the cycles of $\mathcal{FC}(\mathcal{T})$. We show that this encoding always exists.
3 We show that sensor measurements in $\mathcal{K}(c)$ decouple from one cycle to another under single edge exchanges.
4 We use the independence of $\mathcal{K}(c)$ and decoupling of single edge exchange measurements in c to show inductively that no multiple edge exchanges of any size exist where $\Delta \mathbf{s} = 0$, if $E(G) \setminus E(\mathcal{M})$ is a spanning tree.

1) Cycle-Measurement Map: The cycle-measurement map encodes which measurements lie on the edges of a particular cycle. The map $\mathcal{K}(c)$ is defined according to cycles $c \in \mathcal{FC}$, for any arbitrary $\mathcal{FC}$ in G.

Definition 2: With respect to some $\mathcal{FC}$, a cycle-measurement map is $\mathcal{K} : c \to \mathcal{M}$ for all $c \in \mathcal{FC}$ where $s_k \in \mathcal{K}(c)$ if s_k is on an edge in c.
We can also write it as $\mathcal{K}(c) = c \cap E(\mathcal{M})$, though this is an abuse of notation.

For the fundamental cycles associated with the tree in Figure 7(a), 7(b) we construct the following map shown in Table III. We now aim to develop some useful properties of this mapping function.

Consider placement $\mathcal{M}$, and constructed tree $\mathcal{T} = E(G) \setminus E(\mathcal{M})$. We must have $\mathcal{K}(c_k) = \{s_k\}$ where c_k is the k^{th} cycle in $\mathcal{FC}(\mathcal{T})$. This is obvious by construction.

We denote λ_k to be the k^{th} cycle in $\mathcal{FC}(E(G) \setminus E(\mathcal{M}))$ ($\mathcal{FC}_{\mathcal{M}}$ shorthand). From before, $\mathcal{K}(\lambda_k) = \{s_k\}$.

We see that by looking at $\mathcal{K}(c)$, for any cycle in an arbitrary $\mathcal{FC}$, we can encode the cycles construction using basis $\mathcal{FC}_{\mathcal{M}} = \{\lambda_1, \ldots, \lambda_\mu\}$.

Lemma 2: If $E(G) \setminus E(\mathcal{M})$ forms a spanning tree, then for any $\mathcal{FC}$ and $c \in \mathcal{FC}$, $c = \bigoplus_{k:s_k \in \mathcal{K}(c)} \lambda_k$.

Proof: Any cycle c can be represented as a combination of cycles $\lambda \in \mathcal{FC}_{\mathcal{M}}$ since the $\mathcal{FC}_{\mathcal{M}}$ is a cycle basis. If any other λ' outside of the set $\{\lambda_k : s_k \in \mathcal{K}(c)\}$ is used to construct c then $\mathcal{K}(c)$ will include edge containing a s'. Conversely if any additional λ' is needed to construct c, it's s must be in $\mathcal{K}(c)$. ∎

Now we can prove a general case of 'independence' between any two subsets of cycles and the measurements that are placed on them.

Lemma 3: If $E(G) \setminus E(\mathcal{M})$ forms a spanning tree, then for any $\mathcal{FC}$ and subsets $A \neq B$ of cycles in $\mathcal{FC}$ we must have that:

$$\underbrace{\bigcup_{k \in A} \mathcal{K}(c_k)}_{\mathcal{K}_A} \neq \underbrace{\bigcup_{k \in B} \mathcal{K}(c_k)}_{\mathcal{K}_B}. \tag{29}$$

Proof: Suppose that there exists some $\mathcal{FC}$ and partitions A, B where the terms $\mathcal{K}_A$ and $\mathcal{K}_B$ are equal. Since $\mathcal{K}_A$ and $\mathcal{K}_B$ encode some cycle we have that $c_A = c_B$. However, since c_A and c_B are definition fundamental cycle's in $\mathcal{FC}$ there will exist an edge in c_A that is not in c_B thus $c_A \neq c_B$. ∎

To see an example of this consider the cycle-measurement map generated by $\mathcal{FC}(\mathcal{T})$ in Figure 7(a) in Table III. Notice that we cannot construct any partitions A, B where all the covered measurements are equal. For example if $A = \{c_1, c_2\}$ and $B = \{c_3, c_4\}$, we have that $\bigcup_{k \in A} \mathcal{K}(c_k) = \{s_1, s_2\}$ and $\bigcup_{k \in B} \mathcal{K}(c_k) = \{s_1, s_3, s_4\}$.

Notice that the cycle-measurement-map $\mathcal{K}$ constructed for the network in Figure 2(b) does not satisfy that $E(G) \setminus E(\mathcal{M})$ is a spanning tree. In this case we have that $\mathcal{K}(c_1) = \{s_1\}$, $\mathcal{K}(c_2) = \{s_1, s_2\}$, $\mathcal{K}(c_3) = \{s_2\}$ and $\mathcal{K}(c_4) = \{s_1, s_2\}$ for the same cycles. The cycle partition $A = \{c_1\}$, $B = \{c_4\}$ clearly leads to the independence property not holding.

This result leads to the following equivalent results which are used in Section B4.

Corollary 2: A special case is that $\forall c, A \subset \mathcal{FC}$, $\exists s \in \mathcal{K}(c)$ s.t. $s \notin \bigcup_{k \in A} \mathcal{K}(c)$, for any A not including c.

Remark 2: The subset $C \subset \mathcal{FC}$, where $|C| = N$ will have at least N unique sensors.

Remark 3: The indicator vector associated with each $\mathcal{K}(c)$ are a set of linearly independent vectors.

2) Edge Exchange Operator: We encode the transition between any any two spanning trees and show that any such transition can be represented as a set of single edge changes. To motivate this, consider the trees $\mathcal{T}$ and $\mathcal{T}'$ in Figure 7(a), 7(b). The removed edges from G in each tree are $E_R = \{e_6, e_{12}, e_9, e_5\}$ and $E'_R = \{e_7, e_{13}, e_9, e_5\}$. In both trees, e_9 and e_5 do not change.

The main question we want to answer is how to encode the transition between trees by single edge exchanges. Namely, if we define a ΔE operation, do we encode as $\Delta E = \{e_6 \to e_7, e_{12} \to e_{13}\}$ or $\Delta E = \{e_6 \to e_{13}, e_{12} \to e_7\}$.

TABLE IV
FUNDAMENTAL CYCLE BASIS FROM $\mathcal{T}$, $\mathcal{T}'$ AND CYCLE-MEASUREMENT MAP $\mathcal{K}(c)$

$\mathcal{FC}(\mathcal{T})$	$c \cap E_R$	$c \cap E_R$
$c_1 = \{e_6, e_2, e_1, e_7\}$	$\{e_6\}$	$\{e_7\}$
$c_2 = \{e_{12}, e_{13}, e_8, e_4, e_1, e_7\}$	$\{e_{12}\}$	$\{e_7, e_{13}\}$
$c_3 = \{e_5, e_3, e_4, e_8\}$	$\{e_9\}$	$\{e_9\}$
$c_4 = \{e_9, e_8, e_4, e_1, e_7, e_{11}, e_{10}\}$	$\{e_5\}$	$\{e_5\}$

This can be resolved if we look at edge exchanges with respect to $\mathcal{FC}(\mathcal{T})$, as shown in Table IV. Column 1 repeats the cycles in $\mathcal{FC}(\mathcal{T})$. Column 2 maps edges in E_R onto $\mathcal{FC}(\mathcal{T})$ and Column 3 maps edges in E'_R onto $\mathcal{FC}(\mathcal{T})$. This can be seen as an identical mapping function as $\mathcal{K}(c)$ in Section B1, except we replace edges with measurements on them, with edges in E_R. Notice that we can now define a cycle by cycle set of edge exchanges, that define the transition from $\mathcal{T} \to \mathcal{T}'$. So on cycle c_1 we have $\Delta e = (e_6 \to e_7)$ and on cycle c_2 we have $\Delta e = (e_{12} \to e_{13})$.

Definition 3: An edge exchange with respect to $\mathcal{FC}$ is $\Delta E = \{\Delta e_1, \ldots, \Delta e_\mu\}$ where $\Delta e_k = (e_k \to e'_k)$, $e_k \in E_R$, $e'_k \in E'_R$ and $e_k, e'_k \in c_k$.

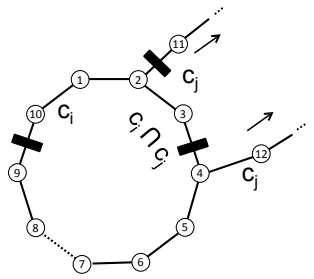
We can generate an edge exchange encoding as follows. First assign e_k the edge in E_R used to construct $c_k \in \mathcal{FC}(\mathcal{T})$. For e'_k use the following procedure:

1) Starting at c_1, set e'_1 to be any element in $c \cap E'_R$.
2) For the k^{th} c_k, set e'_k to be any element in $c_k \cap E'_R$, that has not already been assigned to previous $e'_1 \ldots, e'_{k-1}$.

Lemma 4: For any two $\mathcal{T}, \mathcal{T}'$, at least one edge exchange procedure always exists.

Proof: The set of edges $c \cap E_R$ are equivalent to the mapping $\mathcal{K}(c)$. Therefore, Corollary 2, holds for each incremental cycle to be processed: that is, every additional $c \cap E_R$ set will have a one new edge that can be assigned to e'_k. ∎

Therefore between any two spanning trees there is a well defined set of single edge exchanges performed on the cycles of $\mathcal{FC}(T)$ which encode any arbitrary $\mathcal{T} \to \mathcal{T}'$.



(a)

Fig. 8. 8(a) Illustrative example to see properties in Lemma 5.

3) Decoupling of Measurement along Cycle: We show that when an edge exchange occurs on a cycle, we need to only consider changes of flow values in $\mathcal{K}(c)$. In the development $\mathcal{K}(c)$ and edge exchanges are focused on cycles in $\mathcal{FC}$.

Proposition 2: A single edge exchange on c, with vertices $\{v_0, \ldots, v_m\}$ results in a permutation of an uninterrupted path of the vertices. Therefore, if $e \neq e'$, $p(c, e) \neq p(c, e')$.

Lemma 5: Consider a single edge exchange on c, $e \to e'$ where the following holds:

P1 $\forall s \notin \mathcal{K}(c)$, $\Delta s = 0$,
P2 $\forall s \in \mathcal{K}(c)$, $\Delta s \neq 0$.

Proof: (1) From Proposition 2, any sensor that measured a single vertex in p will measure all the vertices in p before and after the edge exchange. (2) From Proposition 2 since we rearrange all the nodes yet keep a fixed edge to measure flows, all sensors in $\mathcal{K}(c)$ will change values. ∎

Consider the example in Figure 8(a), where the cycle in consideration c_i will have a single edge exchange: for example $\Delta e = (8, 7) \to (1, 2)$ where all $x_i = 1$. In this case, $s = 2$ and and $s' = -2$. Since we assume that the magnitude and direction of the flow is measured, therefore there is no ambiguity. Any other edge will again lead to a new measured value. Since nodes $\{v_0, \ldots, v_{10}\}$ are always connected, all sensors outside of $\mathcal{K}(c)$, for example a measurement on edge $e = (2, 10)$ will not detect an edge exchange.

Remark 4: In condition (2) of Lemma 5, we have $\Delta s \neq 0$ since we measure both magnitude and direction of flow.

4) Inductive Proof: Note our effort is to show sufficiency, for necessity we only need to consider the cycle c in Figure 8(a) If no sensor exists on c_i, $E(G) \setminus E(\mathcal{M})$ is not a spanning tree and every edge exchange leads on c_i will lead $\Delta s = 0$.

From Lemma 4, we can encode any $\mathcal{T} \to \mathcal{T}'$ transition as a set of cycle based edge exchanges. *We now show inductively that no edge exchange exists which leads to* $\Delta s = 0$. So $\mathcal{T} \to \mathcal{T}'$ always leads to $\Delta s \neq 0$.

We show that given any number of non-trivial edge exchanges where at least one s_i s.t. $\Delta s_i \neq 0$.

Base Case Assume $c \subset \mathcal{FC}$, contain a non-trivial edge exchange. From Lemma 5 (P2), $\forall s \in \mathcal{K}$, $\Delta s \neq 0$.

Inductive Hypothesis Assume multiple cycles $C \subset \mathcal{FC}$, contain a non-trivial edge exchange, where $|C| = N$. Assume there exists at least one $s_i \in \mathcal{K}_C(c)$, where $\Delta s_i \neq 0$.

Inductive Step Suppose we find a cycle $c_{n+1} \in \mathcal{FC} \setminus C$ where an edge exchange leads to $\Delta s_i = 0$. From Corollary 3 subset $C \cup c_{n+1}$ will at least $N + 1$ unique sensors and the new cycle must introduce some sensor $s_j \notin \mathcal{K}_C$. From Lemma 5 (2) $\Delta s_j \neq 0$.

C. Proof of Lemma 1

Proof: The matrix $B \in \{-1, 0, +1\}^{N \times M}$ and a valid placement of of size $\mu(G) = M - N + 1$, therefore, $B_N \in \{-1, 0, +1\}^{N \times N-1}$. Since the edges of $G \setminus \mathcal{M}$ maintain a spanning tree property, the graph has 1 connected component. Therefore the incidence matrix must be of rank of $N-1$ [22]. Finally, the matrix B^r_N is a square matrix of size $N - 1$ and invertible. ∎

D. MAP Detector Structure

Here we show how the general MAP detector rule can be evaluated for where we combine edge flows $\mathbf{s}$, load forecasts $\hat{\mathbf{x}}$ and candidate spanning tree $\mathcal{T}$.

$$\hat{\mathcal{T}} = \arg\max_{\mathcal{T} \in \mathbb{T}} \Pr(\mathcal{T} \mid \mathbf{s},\ \hat{\mathbf{x}}) \tag{30}$$

$$= \arg\max_{\mathcal{T} \in \mathbb{T}} \frac{\Pr(\mathbf{s},\ \hat{\mathbf{x}} \mid \mathcal{T}) \Pr(\mathcal{T})}{\Pr(\mathbf{s},\ \hat{\mathbf{x}})} \tag{31}$$

$$= \arg\max_{\mathcal{T} \in \mathbb{T}} \Pr(\mathbf{s},\ \hat{\mathbf{x}} \mid \mathcal{T}) \Pr(\mathcal{T}) \tag{32}$$

$$= \arg\max_{\mathcal{T} \in \mathbb{T}} \Pr(\mathbf{s} \mid \hat{\mathbf{x}},\ \mathcal{T}) \Pr(\hat{\mathbf{x}} \mid \mathcal{T}) \Pr(\mathcal{T}) \tag{33}$$

$$= \arg\max_{\mathcal{T} \in \mathbb{T}} \Pr(\mathbf{s} \mid \hat{\mathbf{x}},\ \mathcal{T}) \Pr(\hat{\mathbf{x}}) \Pr(\mathcal{T}) \tag{34}$$

$$= \arg\max_{\mathcal{T} \in \mathbb{T}} \Pr(\mathbf{s} \mid \hat{\mathbf{x}},\ \mathcal{T}) \tag{35}$$

Lines (30) - (33) convert the MAP detector to a likelihood detector with prior weights. Line (34) conditions on the load forecast $\hat{\mathbf{x}}$. Since $\hat{\mathbf{x}}$ does not depend on the outage hypothesis, (only $\mathbf{s}$ does), the term can be removed leading to (35). Additionally, we assume a uniform prior over all hypotheses, however this does not have to be the case.

E. Proof of Theorem 2

Proof: We need to show that eq. (14) encodes the correct spanning tree: so $\mathbf{f}^\star = \mathbf{f}_{\mathcal{T}_1}$. This can be done by

contradiction: Assume that the solution to $\mathbf{f}^\star = \mathbf{f}_{\mathcal{T}_2}$ represents some other spanning tree or even connected graph with some flow. This implies $(B_N^{r,-1})(\mathbf{x} - B_M^r \mathbf{s}(\mathcal{T}_2, \mathbf{x})) = (B_N^r)^{-1}(\mathbf{x} - B_M^r \mathbf{s}(\mathcal{T}_1, \mathbf{x}))$. Since, $(B_N^{r,-1})$ exists, this implies that $B_M^r \mathbf{s}(\mathcal{T}_2, \mathbf{x}) = B_M^r \mathbf{s}(\mathcal{T}_1, \mathbf{x})$. Which implies that $\mathbf{s}(\mathcal{T}_1, \mathbf{x}) = \mathbf{s}(\mathcal{T}_2, \mathbf{x})$. However from Theorem 1, this is a contradiction.

Finally, we must show that no subgraph of G' is uniquely distinguished from $\mathcal{T}_1$. Such a subgraph occurs for any spanning tree, where a removed edge of the co-tree is added to $\mathcal{T}$. In such a case $\mathbf{s}(\mathcal{T}_1, \mathbf{x}) \neq \mathbf{s}(G', \mathbf{x})$ because:

1) Repeat Lemma 5 for not only edge exchanges, but the case where the cycle has no removed edge. In this case, we have again that [P1] and [P2] hold.
2) The remaining proof can be repeated for not only all spanning trees, but subgraphs formed by these subgraphs formed by co-tree edge addition.

■

F. Proof of Theorem 3

Proof: We construct this test statistic

$$B_H(\mathbf{f}_o - \mathbf{f}_{\mathcal{T}}) = [\mathbf{f}_o(i_1), \ldots, \mathbf{f}_o(i_\mu)]^T \tag{36}$$

Where the matrix $B_H \in \{0,1\}^{|\mu| \times |E|}$ where $B_H(k, i_k) = 1$ for each edge in the co-tree of the particular spanning tree under hypothesis.

The test statistic can be reduced to the following:

$$\begin{aligned} B_H(\mathbf{f}_o - \mathbf{f}_{\mathcal{T}}) &= B_H\left((B_N^{r,-1})\hat{\mathbf{x}} - (B_N^{r,-1})B_M^r \mathbf{s}\right) && (37)\\ &\quad - B_H\left((B_N^{r,-1})\hat{\mathbf{x}} - (B_N^{r,-1})B_M^r \mathbf{s}(\mathcal{T}, \hat{\mathbf{x}})\right) && (38)\\ &= \underbrace{B_H(B_N^{r,-1})B_M^r}_{H}(\mathbf{s} - \mathbf{s}(\mathcal{T}, \hat{\mathbf{x}})) && (39) \end{aligned}$$

Therefore, the new hypothesis test using $x(\mathcal{T})$, is equivalent to the previous test using observed flow with some linear transformation H. The matrix is H is full rank in the $|\mathcal{M}| = \mu(G)$ case. Clearly if H is not square $\mathcal{M} > \mu(G)$, H will not be full rank and the two tests will no longer be the same. ■